Indian Journal of Science and Technology, *Vol 9(41), DOI: 10.17485/ijst/2016/v9i41/90175, November 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Global Convergence Analysis of a Nonlinear Conjugate Gradient Method for Unconstrained Optimization Problems

Ibrahim Abdullahi [1,2*] and Rohanin Ahmad[1]

[1]Department of Mathematical Sciences, Universiti Teknologi Malaysia (UTM), 81310 Johor Bahru, Johor, Malaysia; ibrahim.abdullahi@fud.edu.ng
[2]Department of Mathematics, Federal University Dutse (FUD), P.M.B 7156 Dutse JIgawa State, Nigeria

Abstract

Background/Objectives: The Conjugate Gradient (CG) methods are the well-known iterative methods use for finding solutions to nonlinear system equations. There is need to address the jamming phenomenal facing the current class of this methods. **Methods/Statistical Analysis**: In order to address the shortcomings, we work on the denominator of the Yao et al., CG method which is known to generate descent direction for objective functions by proposing an entire different CG coefficient which can easily switch in case jamming occurs by imposing some parameters thereby guarantee global convergence. **Findings:** The proposed CG formula performs better than classical methods as well as Yao et al. Under Wolfe line search condition, the convergence analysis of the proposed CG formula was established. Some benchmark problems from cute collections are used as basis of strength comparisons of the proposed formula against some other CG formulas. Effectiveness and efficiency of the obtained results for the proposed formula is clearly shown by adopting the performance profile of Dolan and More' which is one of most acceptable techniques of strength comparisons among methods. **Application:** Mathematicians and Engineers who are interested in finding solutions to large scale nonlinear equations can apply the method leading to global optimization dealing with best possible solutions ever for given problems.

Keywords: Conjugate Gradient, Descent Algorithm, Global Convergence, Line Search, Unconstrained Optimization

1. Introduction

In finding solutions to large scale nonlinear unconstrained optimization problems, Conjugate Gradient (CG) methods are among well-known techniques that can handle such class of the problems because of their attractive features, such as low memory requirements together with global convergence properties. The hessian matrix computation which makes it difficult in the computation of step length $\alpha_k > 0$ and CG coefficient β_k whereby at each iteration, there is need to evaluate the hessian matrix for general nonlinear objective function except for quadratic function where hessian matrix for all the iteration is constant, had been taken care of by numerical line search together with objective function evaluation and gradient evaluation at each iteration. With this development there is no need to store hessian matrix rather concern ourselves with objective functions and gradient evaluations. CG methods application set across many field of endeavours such as Engineering, Management science, Operations research among others. For instance, if the parameters from a company has been used to model an unconstrained optimization problem, then CG algorithms can be applied to a find the ideal feasible solution which can be interpret in order to make a decision for the company. The work of[1] focused on the approach to solve symmetric, positive-definite linear systems. However, the method presented by[2] was considered as the first nonlinear CG method.

Let the function $f: \mathbf{R}^n \rightarrow \mathbf{R}$ be continuously differentiable. Given the following unconstrained optimization problem

**Author for correspondence*

$$min\ \{f(x): x \in \mathbf{R}^n\} \tag{1}$$

and g(x) is the gradient of the objective function $f(x)$. Solution to Equation (1), given an initial guess $x_o \in \mathbf{R}^n$, $\{x_k\}$ is the sequence generated CG method:

$$x_{k+1} = x_k + \alpha_k d_k \tag{2}$$

and the direction d_k is defined by

$$d_{k+1} = \begin{cases} g_k & \text{if } k = 0 \\ -g_{k+1} + \beta_k d_k & \text{if } k \geq 1, \end{cases} \tag{3}$$

where x_k, β_k is the current iterate and CG coefficient respectively and α_k>0 is the step-length obtained by a line search. In this paper, we compute α_k using inexact line search given as follows:

$$f(x_k + \alpha_k d_k) \leq \delta \alpha_k g_k^T d_k \tag{4}$$

$$|g(x_k + \alpha_k d_k)^T d_k| \leq \sigma |g_k^T d_k| \tag{5}$$

where d_k is the descent direction and $0 < \delta < \sigma < 1$. Given the wide acceptability of CG methods, several research efforts have been concentrated towards this area with emphasis on the CG coefficient and search direction, to come up with more effective and efficient methods. The pioneer methods are Fletcher-Reeves (FR) method[2], Conjugate Descent (CD) method[3], Dai-Yuan (DY) metho[4], Polak-Rebiere-Polyak (PRP) method[5], Liu-Storey (LS) method[6] and Hestenes-Stiefel (HS) method[1]. In recent years, a variety of CG formulas were given, majorly, differences are in the parameter β_k, the work by[7] discussed details on some CG methods with special emphasis on their global convergence. Table 1 show the summary of the pioneer CG methods.

Table 1. The pioneer CG coefficients (β_k)

No.	β_k	Method	References
1	$\frac{\|\|g_{k+1}\|\|^2}{\|\|g_k\|\|^2}$	Fletcher-Reeves (FR) method	2
2	$-\frac{\|\|g_{k+1}\|\|^2}{d_k^T g_k}$	Conjugate Descent (CD) method	3
3	$\frac{\|\|g_{k+1}\|\|^2}{d_k^T y_k}$	Dai-Yuan (DY) method	4
4	$\frac{g_{k+1} y_k}{\|\|g_k\|\|^2}$	Polak-Rebiere-Polyak (PRP) method	5
5	$-\frac{g_{k+1}^T y_k}{d_k^T g_k}$	Liu-Storey (LS) method	6
6	$\frac{g_{k+1}^T y_k}{d_k^T y_k}$	Hestenes-Stiefel (HS) method	1

where ||.|| denotes the Euclidean norm. The methods in Table 1 behave exactly the same for quadratic function problems when line search is exact and therefore Equation (2) and Equation (3) can be regarded as the linear CG method, otherwise, Equation (2) and Equation (3) is called nonlinear CG method. Usually the search directions satisfy the conjugacy condition $d^T_i\ H\ d_j = 0$, $i \neq j$ where H is the +ve-definite matrix for linear CG. In the case nonlinear CG methods, the conjugacy condition is not satisfied since the hessian $\nabla^2 f(x)$ vary at different points.

Methods such as FR, CD and DY are known for their strong global convergence properties but in the computations, these methods perform poorly. For example, FR method possesses strong convergence properties but computational wise, it performs below PRP and HS as analysed by[8] where he showed the numerical weakness of the FR method. However, methods such as PRP and HS perform better numerically and are known to be among the most efficient methods because of their restart capabilities if it encounters bad direction. A counter example was given by[9] which show that PRP and HS methods may not always converge for general objective functions. Since both categories of the methods, that is, one that has strong global convergence but in practical, perform worse and the other which has good numerical performance but their global convergence is not always guarantee. Research efforts has been ongoing for decades to improve on the existing methods. In line with this, our aim is to propose a CG coefficient that not only converge globally but also have better numerical performance in practice. In recent times, research carried out by[7,10–17] focused on some modified CG methods. Inspired by the works of[18–20] to propose modified CG method called Modified Dai-Yuan (MDY) whose aim was to improve the numerical performance of DY method while retaining its good property of global convergence. By extension, he did same to FR method and called Modified Fletcher-Reeves (MFR), where the parameters β_k were given by

$$\beta_k^{MDY} = \frac{||g_{k+1}||^2}{max\{d_k^T y_k, \mu|g_{k+1}^T d_k|\}} \tag{6}$$

and

$$\beta_k^{MFR} = \frac{||g_{k+1}||^2}{max\{||g_k||^2, \mu|g_{k+1}^T d_k|\}}, \tag{7}$$

where $y_k = g_{k+1} - g_k$ and $\mu > 1$. In[21] applied the idea of the by[22] to the HS method by proposing a CG method:

$$\beta_k^{YWH} = \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||} g_k^T g_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} \quad (8)$$

under strong Wolfe line search with parameter σ $\frac{1}{3}$. For general objective functions, the YWH method was known to be globally convergent and can produce sufficient descent directions.

In contrast to some existing modified methods and in particular the method from[21], based on the work by[21], we propose a new modified CG method given as

$$\beta_k^{IR1} = \begin{cases} \beta_k^{New}, \ if \ \left|1 - \frac{g_k^T g_{k-1}}{||g_{k-1}||||g_k||}\right| < \mu, \\ 0 \ otherwise. \end{cases} \quad (9)$$

where

$$\beta_k^{New} = \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||} g_k^T g_{k-1}}{\mu |g_k d_{k-1}| + d_{k-1}^T (g_k - g_{k-1})} \quad (10)$$

The work of[23–27] focused on other related areas such as gradient orientation, mathematics test with Rasch model, applying learning analytics in mathematics, simulation of real time nonlinear process and integro-differential equations.

The remaining parts of the paper are in the order. In Section 2, we present the algorithm and show that our corresponding formula can always guarantee descent condition. In Section 3, convergence analysis for the proposed method is presented. Section 4 entails the proposed method's numerical results and also the representation of proposed method against some CG methods using Dolan and More's performance profile[28] and lastly the conclusion.

2. Algorithm and Descent Property

In this Section, we describe the CG algorithm and show that the propose formula (Equation (9)) possesses the descent properties.

2.1 Algorithm

1. Initialization. Given constants $\epsilon > 0$, $\delta \in (0,1)$, $\sigma \in (\delta, 1)$ μ >1, select $x_o \in \mathbf{R}^n$, set $k = 0$, $d_o = -g_o$.
2. Test for convergence. If $||g_k|| \le \epsilon$, then stop. Otherwise go to 3.
3. Compute α_k based on inexact line search Equation (4) and Equation (5).
4. Variable update, $x_{k+1} = x_k + \alpha_k d_k$. Compute $f(x_{k+1})$ and g_{k+1}.
5. Computation of parameter β_k based on Equation (9) and Equation (10).
6. Generate d_k using Equation (3). Set $k = k + 1$ and go to 2.

Lemma 1. Let the sequences $\{x_k\}$ and $\{d_k\}$ be generated by the Algorithm 2.1 for β_k^{IR1}. Then, $g_k^T d_k < 0$ holds true.

Proof. We proceed by induction to arrive at the conclusion. It is obvious to have $g_1^T d_1 = -||g_1||^2 < 0$, if $k = 1$. Assume that $g_{k-1}^T d_{k-1} < 0$ holds true for $k - 1$, to obtain

$g_k^T d_k < 0$ particularly for our method (β_k^{IR1}).

From the search direction, we have

$$g_k^T d_k = -||g_k||^2 + \beta_k g_k^T d_{k-1} \le -||g_k||^2 + |\beta_k||g_k^T d_{k-1}| \quad (11)$$

Case (i) If $\beta_k^{IR1} = 0$. It follows clearly from Equation (2), $g_k^T d_k \le -||g_k||^2 < 0$. Case (ii) If $\beta_k^{IR1} = \beta_k^{New}$. Recall from Wolfe line search,

$d^T k-1 (g_k - g_{k-1}) = d^T k-1\, g\, k - d^T k-1\, g\, k-1 \ge \sigma d^T k-1\, g\, k-1 - d^T k-1\, g\, k-1 = (\sigma-1)\, d^T k-1\, g\, k-1 > 0.$ (12)

It follows from Equation (10)

$$g_k^T d_k = -||g_k||^2 + \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||} g_k^T g_{k-1}}{\mu |g_k d_{k-1}| + d_{k-1}^T (g_k - g_{k-1})} . g_k^T d_{k-1}$$

$$\le -||g_k||^2 + \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||} g_k^T g_{k-1}}{\mu |g_k d_{k-1}| + d_{k-1}^T (g_k - g_{k-1})} . |g_k^T d_{k-1}|$$

$$\le -||g_k||^2 + \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||} g_k^T g_{k-1}}{\mu |g_k d_{k-1}|} . |g_k^T d_{k-1}|. \quad (13)$$

Note that $\beta_k^{IR1} \ne 0$ and $g_k^T g_{k-1} > 0$, we have $0 < cos\theta_k < 1$ and let θ_k be the angle between g_k and g_{k-1}

$$= -||g_k||^2 + \frac{||g_k||^2 - ||g_k||^2 cos\theta_k}{\mu} = \frac{-(\mu - 1 + cos\theta_k)||g_k||^2}{\mu} < 0$$

For $\mu \ge 1$ then $g_k^T d_k < 0$ holds for all $k \ge 1$.

Lemma 2. *The relation* $0 \le \beta_k^{IR1} \le \frac{g_k^T d_k}{g_{k-1}^T d_{k-1}}$ *holds for any* $k \ge 1$.

Proof. From Eq. 9,

$$0 \le \beta_k^{IR1} \le \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||} g_k^T g_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} = \beta_k^{YWH}.$$

If $\beta_k = \beta_k^{YWH}$, it follows from Eq. 3,

$$g_k^T d_k = -||g_k||^2 + \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||} g_k^T g_{k-1} . g_k^T d_{k-1}}{d_{k-1}^T (g_k - g_{k-1})}.$$

Since $\beta_k^{YWH} \neq 0$, then $g_k^T g_{k-1} > 0$, we have $0 < \cos\theta_k < 1$, where θ_k is the angle between g_k and g_{k-1}.

From Eq. 3 and Eq. 8, we have

$$g_k^T d_k = -||g_k||^2 + \frac{||g_k||^2 g_{k-1} d_{k-1} - ||g_k||^2 \cos\theta_k g_k d_{k-1}}{d_{k-1}^T (g_k - g_{k-1})}$$

$$< \frac{||g_k||^2 g_{k-1} d_{k-1} - ||g_k||^2 \cos\theta_k g_{k-1}^T d_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} = \frac{(1-\cos\theta_k)||g_k||^2 g_{k-1}^T d_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} < 0 \tag{14}$$

Therefore, from Eq. 14 we have

$$\beta_k^{IR1} \leq \frac{||g_k||^2 - \frac{||g_k||}{||g_{k-1}||} g_k^T g_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} = \frac{(1-\cos\theta_k)||g_k||^2}{d_{k-1}^T (g_k - g_{k-1})} < \frac{g_k^T d_k}{g_{k-1}^T d_{k-1}} \tag{15}$$

Thus, the proof is completed.

3. Global Convergence

In discussing the global convergence of the proposed, some basic assumptions on the objective functions are necessary.

Assumption (3.1):

i. A given objective function $f(x)$ is bounded below on the level set $\Omega = \{x \in \mathbf{R}^n : f(x) \leq f(x_0)\}$ and x_0 is the initial point.

ii. In some neighbourhood $\mathbf{P}$ of $\mathbf{\Omega}$, the objective function $f(x)$ is continuously differentiable and its $g(x) = \nabla f(x)$ satisfies Lipschitz condition, namely, $\exists$ a constant $L > 0$:

$$||g(x) - g(y)|| \leq L||x - y||, \forall\, x,y \in \mathbf{P}. \tag{16}$$

From the above assumptions on the objective function $f(x)$, $\exists$ a constant $\gamma \geq 0$:

$$||\nabla f(x)|| \leq \gamma, \forall\, x \in \mathbf{\Omega}. \tag{17}$$

To prove the global convergence of the proposed methods, the result of the following lemma, usually called Zoutendijk condition is required. For proof, refer to[29,30].

Lemma 3. Supposed **Assumption** (3.1) holds and consider any CG method of the form $x_{k+1} = x_k + \alpha_k d_k$ and the direction $d_{k+1} = -g_{k+1} + \beta_k^{IR1} d_k, d_0 = -g_0$ where α_k satisfies Equation (4) and Equation (5). Then,

$$\sum_{k \geq 1} \frac{(g_k^T d_k)^2}{||d_k||^2} < +\infty. \tag{18}$$

From Lemma 3, we have the following theorem which presents the global convergence of the proposed method:

Theorem 4. *Let* **Assumption** (3.1) *holds* and the sequence $\{x_k\}$ and $\{d_k\}$ be generated by Algorithm 2.1 with β_k^{IR1}, α_k is obtained by Equation (4) and Equation (5). Then

$$\liminf_{k \to \infty} || g_k || = 0. \tag{19}$$

Proof. Proceed using contradiction to arrive at the conclusion. Suppose that $\liminf_{k \to \infty} || g_k || \neq 0$, it implies that $\exists\, m > 0$ such that

$$||g_k|| \geq m, \forall\, k \geq 0. \tag{20}$$

From Eq. 3, we have

$$(\beta k^{IR1}\, dk{-}1)^2 = (dk{-}1 + gk)^2, \tag{21}$$

it follows from Eq. 21 and Lemma 3.

$$||d_k||^2 = (\beta_k^{IR1})^2 ||d_{k-1}||^2 - 2g_k^T d_k - ||g_k||^2 \leq (\frac{g_k^T d_k}{g_{k-1}^T d_{k-1}})^2 ||d_{k-1}||^2 - 2g_k^T d_k - ||g_k||^2. \tag{22}$$

Dividing both side of Eq. 22 by $(g_k^T d_k)^2$ to get

$$\frac{||d_k||^2}{(g_k^T d_k)^2} \leq \frac{||d_{k-1}||^2}{(g_{k-1}^T d_{k-1})^2} - \frac{2}{g_k^T d_k} - \frac{||g_k||^2}{(g_k^T d_k)^2} = \frac{||d_{k-1}||^2}{(g_{k-1}^T d_{k-1})^2} - (\frac{1}{||g_k||} + \frac{||g_k||}{g_k^T d_k})^2 + \frac{1}{||g_k||^2}$$

$$\leq \frac{||d_{k-1}||^2}{(g_{k-1}^T d_{k-1})^2} + \frac{1}{||g_k||^2}. \tag{23}$$

Noting that $\frac{||d_1||^2}{(g_1^T d_1)^2} = \frac{1}{||g_1||^2}$, by recurrence formula Eq. 23, we get

$$\frac{||d_k||^2}{(g_k^T d_k)^2} \leq \frac{||d_{k-1}||^2}{(g_{k-1}^T d_{k-1})^2} + \frac{1}{||g_k||^2} \leq \frac{||d_{k-2}||^2}{(g_{k-2}^T d_{k-2})^2} + \frac{1}{||g_{k-1}||^2} + \frac{1}{||g_{k-1}||^2} \leq \cdots \leq \sum_{i=1}^{k} \frac{1}{||g_i||^2}.$$

Hence

$$\frac{||d_k||^2}{(g_k^T d_k)^2} \leq \sum_{i=1}^{k} \frac{1}{||g_i||^2} \leq \frac{k}{m^2}, \tag{24}$$

furthermore,

$$\frac{(g_k^T d_k)^2}{||d_k||^2} \geq m^2 \sum_{k \geq 0} \frac{1}{k} = +\infty. \tag{25}$$

This contradicts Zoutendijk condition in Eq. 18.

4. Numerical Results

The presentation of the simulation results on the test problems for our proposed method where $\beta_k = \beta_k^{IR1}$ against some existing methods in the literature are done in this part. We consider some test problems from[31,32] to validate the numerical strength of our method versus some methods in existence, using inexact line search Conditions (4) and (5) for all methods in this paper for easy comparison where $\delta = 0.0001$ and $\sigma = 0.01$.

The parameters such as number of iterations (it), number of function evaluations (nf) and CPU time (t) were considered to evaluate the computational capability of the proposed method β_k^{IR1} as compared with FR, DY and YWH. For each test problem, the stopping criterion is taken as $||g_k|| \leq \in$, where $\in = 10^{-5}$. We implemented the method using MATLAB R2014 in double precision arithmetic on CP computer with CPU 1.30 GHz and 4.00GB RAM. Tables 2, 3 and 4 show the simulation results of the proposed method against some methods (FR, DY and YWH). The symbol (–) implies failure in numerical computation while (∗) means that number of iterations or function evaluations exceeded the maximum limit set. For iteration, we set 5000 as the maximum while 20000 is the maximum for number of function evaluations.

Table 2. Numerical results of IR1, FR, DY and YWH

Fun./Dim	IR1 it/nf/cpu(s)	FR it/nf/cpu(s)	DY it/nf/cpu(s)	YWH it/nf/cpu(s)
Rosenbrock/2	16/70/0.203	34/114/0.640	23/88/0.187	16/70/0.234
Rosenbrock/100	818/3830/5.741	1084/5641/8.190	1015/5465/8.658	818/3835/5.788
Denschnc/50	21/195/0.250	48/543/0.624	–	19/201/0.234
Denschnc/100	19/186/0.265	48/543/0.608	–	16/174/0.203
Denschnc/800	16/159/0.359	43/510/1.014	–	16/174/0.359
Denschnc/10000	16/159/2.262	39/480/7.410	–	13/147/2.309
Denschnc/100000	13/146/23.463	23/326/51.636	–	10/120/19.204
Denschna/50	12/59/0.094	12/57/0.140	10/50/0.094	11/55/0.078
Denschna/500	11/57/0.109	11/55/0.156	10/50/0.109	11/55/0.109
Denschna/5000	11/57/0.437	11/55/0.359	9/48/0.281	11/55/0.624
Denschna/100000	11/57/6.614	9/51/5.647	8/46/4.898	9/51/5.678
Denschnb/500	7/31/0.047	8/35/0.078	8/35/0.078	8/35/0.078
Denschnb/1000	7/31/0.078	8/35/0.078	8/35/0.094	8/35/0.078
Denschnb/5000	7/31/0.125	8/35/0.156	8/35/0.156	8/35/0.140
Denschnb/15000	6/29/0.265	8/35/0.328	8/35/0.312	8/35/0.296
Denschnb/100000	6/29/2.012	7/33/2.153	7/33/1.888	8/35/2.278
Denschnf/50	10/42/0.125	23/83/0.172	22/79/0.187	10/42/0.109
Denschnf/1000	10/42/0.140	22/81/0.250	22/79/0.265	10/42/0.125
Denschnf/5000	9/40/0.312	22/81/0.530	22/79/0.484	9/40/0.250
Denschnf/18000	8/38/0.640	22/81/1.560	22/79/1.451	8/38/0.686
Denschnf/100000	8/38/5.257	21/79/9.937	21/77/9.251	8/38/4.852
sine/17000	5/15/0.265	21/57/0.983	21/57/0.889	10/29/0.515
sine/100000	1/4/0.281	1/4/0.359	1/4/0.281	1/4/0.296
G Quartic/500	5/15/0.047	5/15/0.078	5/15/0.047	5/15/0.047
G Quartic/1000	5/15/0.047	5/15/0.078	5/15/0.047	5/15/0.047
G Quartic/5000	4/13/0.047	4/13/0.094	4/13/0.047	4/13/0.078
G Quartic/10000	3/11/0.094	3/11/0.109	3/11/0.094	3/11/0.094
G Quartic/100000	2/9/0.796	2/9/0.874	2/9/0.718	2/9/0.780
Emaratos/5000	55/251/0.920	29/191/0.671	29/191/0.686	64/292/1.076
Emaratos/10000	55/251/1.435	29/191/1.061	29/191/0.998	64/292/1.622

Emaratos/100000	55/251/15.663	29/191/11.216	28/189/10.483	64/292/18.595
E Himmelblau/50	7/24/0.062	10/31/0.078	10/31/0.094	8/26/0.062
E Himmelblau/100	7/24/0.062	9/29/0.078	9/29/0.062	8/26/0.062
E Himmelblau/5000	5/20/0.109	8/27/0.172	8/27/0.140	8/26/0.140
E Himmelblau/10000	5/20/0.202	8/27/0.234	8/27/0.218	7/24/0.156
E Himmelblau/24000	5/20/0.312	8/27/0.515	8/27/0.468	7/24/0.406

Table 3. Numerical results of IR1, FR, DY and YWH

Fun./Dim	IR1 it/nf/cpu(s)	FR it/nf/cpu(s)	DY it/nf/cpu(s)	YWH it/nf/cpu(s)
E beale/2	11/36/0.094	24/71/0.172	23/68/0.156	16/50/0.109
E beale/10	11/36/0.078	24/71/0.172	23/68/0.156	15/48/0.109
E beale/50	11/36/0.094	23/69/0.203	23/68/0.156	15/48/0.109
E beale/10000	10/34/0.738	23/69/1.420	22/66/1.326	14/46/0.920
E Whitehoslt/50	6/21/0.047	21/52/0.140	21/52/0.109	11/32/0.094
E Whitehoslt/100	6/21/0.031	18/46/0.109	21/52/0.109	10/30/0.078
E Whitehoslt/500	6/21/0.062	17/44/0.172	17/44/0.140	10/30/0.062
E Whitehoslt/100000	6/21/2.418	9/28/3.151	9/28/3.182	8/26/3.120
E Triadiagonal2/100000	3/10/1.092	3/10/1.108	3/10/1.186	3/10/1.170
E Penalty/2	6/21/0.062	6/21/0.047	7/23/0.047	6/21/0.047
Brown/500	5/15/0.062	5/15/0.125	5/15/0.078	5/15/0.140
Brown/5000	4/13/0.343	5/15/0.437	5/15/0.328	4/13/0.234
Brown/100000	3/11/4.493	3/11/4.805	3/11/4.462	3/11/5.039
Triadiagonalwhl/500	1599/7955/19.765	*	*	1598/7943/20.499
Triadiagonalwhl/1000	3177/15863/59.733	*	*	3177/15883/50.888
Triadiagonalwhl/100000	36/187/39.234	*	62/383/85.457	34/174/36.582
G Fletcher/4	16/68/0.094	24/91/0.187	23/89/0.156	16/68/0.094
G Fletcher/40	25/92/0.172	29/103/0.187	27/100/0.203	25/92/0.172
G Fletcher/500	23/89/0.202	23/92/0.234	22/90/0.203	23/89/0.172
G Fletcher/10000	17/80/0.733	21/97/0.905	21/97/0.952	17/81/0.733
G Fletcher/100000	15/78/8.050	16/84/9.079	16/84/9.017	16/83/8.252
G triadiagonal/100000	1/3/3.666	1/3/3.900	1/3/3.682	1/3/3.666
Raydan2/100	2/5/0.031	2/5/0.031	2/5/0.016	2/6/0.047
Raydan2/500	2/5/0.031	2/5/0.031	2/5/0.047	2/6/0.016
Raydan2/5000	2/5/0.031	2/5/0.031	2/5/0.016	2/6/0.047
Raydan2/6000	2/5/0.047	2/5/0.031	2/5/0.031	2/6/0.047
Raydan1/2	4/13/0.047	4/13/0.047	4/13/0.031	4/13/0.031
G PSC1/50	7/37/0.047	7/35/0.078	7/35/0.078	7/38/0.062
G PSC1/800	7/31/0.094	9/51/0.140	9/51/0.172	8/41/0.109
G PSC1/5000	7/29/0.234	9/42/0.374	9/40/0.312	9/44/0.359
G PSC1/50000	5/19/1.544	9/39/3.635	9/39/3.198	9/40/3.198
E PSC1/5000	6/21/0.125	8/25/0.250	8/25/0.203	6/21/0.156
E QP1/20	6/23/0.047	5/20/0.062	5/20/0.062	7/26/0.047

E QP1/150	6/23/0.031	7/28/0.062	7/28/0.062	8/31/0.078
Tridia/20	19/39/0.094	19/39/0.125	19/39/0.125	19/39/0.125
Tridia/800	465/931/2.683	356/713/2.137	465/931/3.089	465/931/2.714
Tridia/3000	1347/2695/12.293	747/1495/8.002	1347/2695/12.714	1347/2695/9.968

Table 4. Numerical results of IR1, FR, DY and YWH

Fun./Dim	IR1 it/nf/cpu(s)	FR it/nf/cpu(s)	DY it/nf/cpu(s)	YWH it/nf/cpu(s)
Tridia/5000	2044/4089/22.433	861/1723/9.157	2044/4089/20.951	2044/4089/18.190
Tridia/10000	3634/7269/59.358	1641/3283/24.945	3634/7269/56.831	3634/7269/56.550
Arwhead/50	3/14/0.062	3/14/0.062	3/14/0.031	3/14/0.047
Arwhead/500	3/15/0.031	3/15/0.031	3/15/0.062	3/15/0.062
Arwhead/7500	3/18/0.078	3/18/0.078	3/18/0.078	3/18/0.125
Arwhead/100000	2/16/0.827	2/16/0.967	2/16/0.842	2/16/0.905
Nondia/15500	4939/9879/162.272	981/1963/30.561	4939/9879/154.706	4939/9879/155.939
Nondia/100000	219/439/37.019	121/243/20.998	219/439/39.156	219/439/36.535
DQDRTIC/100	5/11/0.047	5/11/0.094	5/11/0.047	5/11/0.016
DQDRTIC/450	5/11/0.047	5/11/0.078	5/11/0.062	5/11/0.047
DQDRTIC/50000	5/11/0.562	5/11/0.250	5/11/0.515	5/11/0.515
Cube/2	20/87/0.125	24/102/0.172	23/90/0.156	20/86/0.140
Cube/700	132/524/1.466	145/661/2.293	84/470/1.388	126/591/2.137
Cube/7000	43/190/3.073	44/226/3.88	43/228/3.510	62/291/4.430
QUARTC/50	2/8/0.016	2/8/0.031	2/8/0.031	2/8/0.031
QUARTC/50000	2/8/0.421	2/8/0.437	2/8/0.374	2/8/0.484
BDEXP/80	1/3/0.016	1/3/0.047	1/3/0.016	1/3/0.001
BDEXP/8000	1/3/0.078	1/3/0.062	1/3/0.078	1/3/0.047
BDEXP/13500	1/3/0.078	1/3/0.125	1/3/0.094	1/3/0.125
cosine/2	2/7/0.031	2/7/0.031	2/7/0.031	2/7/0.016
cosine/4000	4/17/0.078	4/17/0.109	4/17/0.078	4/17/0.094
cosine/70000	3/15/1.123	3/15/1.217	3/15/1.201	3/15/1.217
Diagonal7/30	2/6/0.031	2/6/0.031	2/6/0.016	2/6/0.031
Diagonal7/300	2/6/0.031	2/6/0.047	2/6/0.016	2/6/0.047
Diagonal7/80000	2/6/0.265	2/6/0.312	2/6/0.343	2/6/0.296
Diagonal8/25	2/6/0.001	2/6/0.031	2/6/0.016	2/6/0.016
Diagonal8/250	2/6/0.016	2/6/0.031	2/6/0.031	2/6/0.031
Diagonal8/100000	2/6/0.546	2/6/0.530	2/6/0.484	2/6/0.515
TET/4	7/107/0.094	6/84/0.109	9/157/0.140	5/64/0.062
TET/8	7/105/0.109	6/84/0.094	9/157/0.156	5/64/0.062
TET/4000	9/94/0.452	5/71/0.374	8/146/0.640	4/54/0.250
Power/2	4/37/0.047	4/37/0.062	4/37/0.031	4/37/0.078
Power/25	33/630/0.406	31/615/0.421	48/949/0.733	34/655/0.437
Himmelbg/10	1/5/0.031	1/5/0.062	1/5/0.016	1/5/0.031
Himmelbg/100	1/5/0.016	1/5/0.016	1/5/0.016	1/5/0.001

The performance profiles of Dolan and More'[25] was used to compare the numerical strength of the proposed method against some known CG methods such as FR, DY and YWH methods based on it, fn and t. We plot fraction $p_s(t)$ of the test problems for which the method is within a factor t of the best time for each method. The left hand side of the figures give the % of how fast is a particular method in solving the test problems. The right hand side of the figures give the % of test problems that are successfully solved by each method. The solver with large probability $p_s(t)$ is regarded as the best solver for the test problems.

From Figures 1-3, we see that among these methods, the proposed method (IR1) performs better than FR and DY methods in its entirety for the test problems. Also note, from Figure 1 and 2, IR1 method performs faster

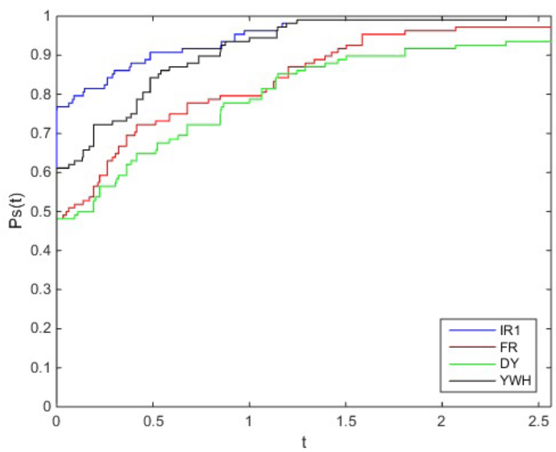

Figure 1. Performance profile based on iteration for IR1 versus FR, DY and YWH

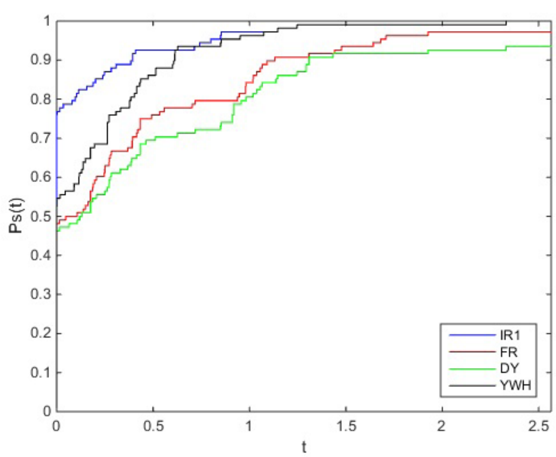

Figure 2. Performance profile based on function evaluation. IR1 versus FR, DY and YWH.

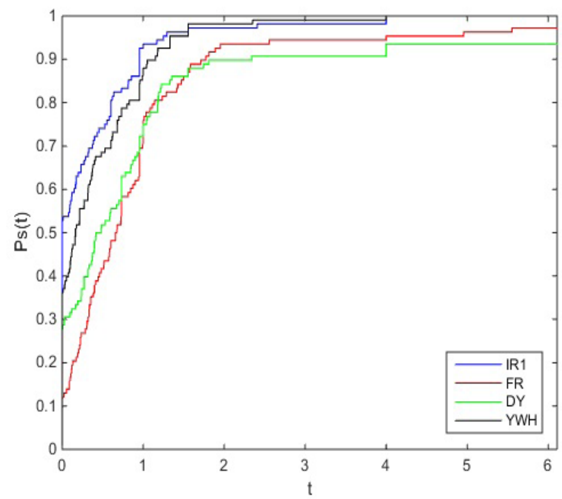

Figure 3. Performance profile based on CPU time. IR1 versus FR, DY and YWH.

than YWH method at first and slightly above towards end, which shows that the two methods were able to solve all the test problems successfully. In Figure 3, the IR1 method obtained optimal solutions for the test problems within shortest time as compared to the execution time for FR and DY and slightly above the YWH method.

5. Conclusion

In this paper, a new type of a modified CG method was proposed for solving unconstrained problems. The proposed method generated descent directions using Wolfe line search condition. Under line search Condition (4) and (5), we established the global convergence of the proposed method. The parameter β_k^{IR1} with $\mu > 1$, specifically we take $\mu = 1.2$ for experiment conducted in this paper. The simulation results of the proposed method shown to be efficient when compared against some CG methods (FR, DY and YWH). We employed one of the best methods of comparison (Performance Profiles by Dalon and More') to show the effectiveness of our proposed methods.

6. Acknowledgement

The authors would like to thank Universiti Teknologi Malaysia through the financial assistance from the grant RUG 14H58 and Malaysian Ministry of Higher Education for the opportunity created to do research.

7. References

1. Hestenes MR, Stiefel E. Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards. 1952; 49:409–36.
2. Fletcher R, Reeves CM. Function minimization by conjugate gradients. The computer Journal. 1964; 7(2):149–54.
3. Fletcher R. Practical methods of optimization. John Wiley and Sons; 2013.
4. Dai YH, Yuan Y. A nonlinear conjugate gradient method with a strong global convergence property. SIAM Journal on Optimization. 1999; 10(1):177–82.
5. Polyak BT. The conjugate gradient method in extremal problems. USSR Computational Mathematics and Mathematical Physics. 1969; 9(4):94–112.
6. Hu Y, Storey C. Global convergence result for conjugate gradient methods. Journal of Optimization Theory and Applications. 1991; 71(2):399–405.
7. Hager WW, Zhang H. A survey of nonlinear conjugate gradient method. Pacific Journal of Optimization. 2006; 2(1):35–58.
8. Powell MJD. Restart procedures for the conjugate gradient method mathematical programming. Springer. 1977; 12(1):241–54.
9. Powell MJD. Nonconvex minimization calculations and the conjugate gradient method. Lecture Notes in Mathematics, Numerical Analysis, Springer. 1984; 1066:122–41.
10. Hager WW, Zhang H. A new conjugate gradient method with guaranteed descent and an efficient line search. JSIAM Journal on Optimization. 2005; 16(1):170–92.
11. Liu J, Feng Y. Global convergence of a modified ls nonlinear conjugate gradient method. Procedia Engineering. 2011; 15:4357–61.
12. Wei Z, Li G, Qi L. New nonlinear conjugate gradient formulas for large-scale unconstrained optimization problems. Applied Mathematics and Computation. 2006; 179(2):407–30.
13. Polyak BT. The conjugate gradient method in extremal problems. USSR Computational Mathematics and Mathematical Physics. 1969; 9(4):94–112.
14. Ibrahim A, Rohanin A. Convergence analysis of a new conjugate gradient method for unconstrained optimization. Applied Mathematical Sciences. 2015; 9(140):6969–84.
15. Zhang Y, Zheng H, Zhang C. Global convergence of a modified PRP conjugate gradient method. Procedia Engineering. 2012; 31:986–95.
16. Du X, Liu J. Global convergence of a spectral hs conjugate gradient method. Procedia Engineering. 2011; 15:1487–92.
17. Mohd R, Abdelrhaman A, Mustafa M, Ismail M. The convergence properties of a new type of conjugate gradient methods. Applied Mathematical Sciences. 2015; 9(54):2671–82.
18. Zhifeng D, Fenghua W. Another improved Wei–Yao–Liu nonlinear conjugate gradient method with sufficient descent property. Applied Mathematics and Computation. 2014; 1(8):33–44.
19. Dai Y. A nonmonotone conjugate gradient algorithm for unconstrained optimization. Journal of Systems Science and Complexity. 2002; 2(15):139–45.
20. Xian-zhen J, Jin-bao J. A sufficient descent Dai–Yuan type nonlinear conjugate gradient method for unconstrained optimization problems. Nonlinear Dynamics. 2013; 72(1):101–12.
21. Yao S, Zengxin W, Hai H. A note about WYLs conjugate gradient method and its applications. Applied Mathematics and Computation. 2007; 2(191):381–8.
22. Zengxin W, Shengwei Y, Liying L. The convergence properties of some new conjugate gradient methods. Applied Mathematics and Computation. 2006; 2(183):1341–50.
23. Vaithiyanathan V, Lakshmi KD, Sundar JAK, Ahmed MI, Sangeetha V, Sivagami R. Evaluation of distance functions for the comparison of gradient orientation histograms. Indian Journal of Science and Technology. 2015 Dec; 8(35). DOI: 10.17485/ijst/2015/v8i35/86808.
24. Khairani AZ, Abd Razak N. Modeling a multiple choice mathematics test with the rasch model. Indian Journal of Science and Technology. 2015 Jun; 8(12). DOI: 10.17485/ijst/2015/v8i12/70650.
25. Jeong-Phil H, In-Ae K, Seung-Soo S. A case study of applying learning analytics in general mathematics class. Indian Journal of Science and Technology. 2015 Sep; 8(21). DOI: 10.17485/ijst/2015/v8i21/78382.
26. Chakravarthi MK, Pannem KV, NithyaV. Design and Simulation of internal model controller for a real time nonlinear process. Indian Journal of Science and Technology. 2015 Aug; 8(18). DOI0: 10.17485/ijst/2015/v8i19/76699.
27. Venkatesh SG, Ayyaswamy SK, Balachandar SR, Kannan K. Wavelet solution for class of nonlinear integro-differential equations. Indian Journal of Science and Technology. 2013 Jun; 6(6). DOI: 10.17485/ijst/2013/v6i6/33919.
28. Elizabeth DD, Jorge JM. Benchmarking optimization software with performance profiles. Mathematical Programming. 2002; 91(2):201–13.
29. Zoutendijk G. Nonlinear programming. Integer and Nonlinear Programming. 1970; 143(1):37–86.
30. Yuhong D, Jiye H, Guanghui L, Defeng S, Hongxia Y, Ya-Xiang Y. Convergence properties of nonlinear conjugate gradient methods. SIAM Journal on Optimization. 2000; 10(2):345–58.
31. Andrei N. An unconstrained optimization test functions collection. Adv Model Optim. 2008; 10(1):147–61.
32. Andrei N. Test functions for unconstrained optimization. Research Institute for informatics. Center for Advanced Modeling and Optimization; 2004. p. 1–10.